710 James Robertson Pkwy
Nashville, TN 37243
Phone: (615) 741-2731

🐦 @TNedu
Ⓕ Tennessee Education
www.TN.gov/Education

CTSO Course Alignments: Programming and Logic I

Below you will find standards for the Programming and Logic I course aligned with competitive events from appropriate career and technical student organizations (CTSOs). Knowing the aligned events for your organization will allow you to have additional tools for teaching course standards, as well as increase student engagement and preparation in your CTSO activities. The final column recommends potential tools from other CTSO organizations. Even if your students are not participating in these organizations, available rubrics, tools, and materials can also add to the instructional resources at your disposal for best teaching your content.

Important to note: While the aligned activities below can be important tools in teaching course standards, it is important to note that events may not cover a standard in its entirety and should not be the sole instructional strategy used to address a standard.

	STANDARD	ALIGNED SkillsUSA & TSA COMPETITIVE EVENTS/PROGRAMS	OTHER POTENTIAL CTSO TOOLS & RESOURCES
1	Using news articles and instructional materials, investigate key milestones in the development of computers and logical devises. Create and present a document and/or illustration depicting the timeline of development that led to modern-day operating systems, programmable controllers, and widespread digital communications via the Internet and wireless networks, citing specific textual evidence. (TN Reading 1, 2; TN Writing 2, 4)	• **TSA**: Essays on Technology, Prepared Presentation	• **FFA:** Extemporaneous Speaking • **HOSA**: Researched Persuasive Writing, Prepared Speaking
2	Compare and contrast the benefits, features, and typical applications of common modern programming languages and environments. Craft an argument to defend the choice of a certain language to solve a particular problem, developing claim(s) and counterclaim(s) with specific textual evidence and reasoning. (TN Reading 1, 2, 4, 5; TN Writing 1, 4)	• **TSA**: Debating Technological Issues	• **FBLA:** Help Desk • **FCCLA**: Advocacy • **HOSA**: Biomedical Debate
3	Using news articles and text of legislation, analyze ethical programming practices, including but not limited to the issues of confidentiality, privacy, piracy, fraud and misuse, liability, copyright, open source software, trade secrets, and sabotage. For example, research and report on the effects of unethical programming practices on a business. (TN Reading 1, 2; TN Writing 7)	• **TSA**: Essays on Technology, Prepared Presentation	• **FBLA:** Business Ethics • **HOSA**: Researched Persuasive Writing, Prepared Speaking

4	Differentiate between system-level and application solutions, and identify an appropriate code-based strategy to solve a given problem. For example, given a file management problem, determine when a command-line script will be more efficient than a high-level program solution. (TN Reading 4, 5; TN Math N-Q, A-SSE, F-IF)		

5	Apply the system management tools present in a programming development environment to:
	    a. Select the most appropriate programming language for the task at hand
	    b. Develop syntactically correct program code using current best practices and emerging classes of development techniques
	    c. Use a compiler to interpret the source code and produce executable program code
	(TN Reading 3, 4, 5; TN Writing 4)		

6	In the process of developing and implementing programming solutions, develop strategies that work within the constraints of major operating system fundamentals, such as:
	    a. Security protocols and procedures for accessing files and folders
	    b. File management syntax requirements, including but not limited to creating, naming, organizing, copying, moving, and deleting files
	    c. File naming conventions, as they apply across multiple software applications and file types.
	(TN Reading 3, 4; TN Writing 6)		

7	Write pseudocode and construct a flowchart for a process before starting to develop the program code. For example, code and flowchart a simple process that takes an integer and report whether it is odd or even. (TN Reading 3, 4; TN Writing 4)		

8	Organize and develop a plan to acquire and manage the data values for a process, including the following:
	    a. Data types, such as string, numeric, character, integer, and date
	    b. Program variable names
	    c. Variables and constants
	    d. Arrays (at least one- and two-dimensional), subscripts
	    e. Input from files and user responses
	    f. Output to files and reports
	(TN Reading 5)		

#			

9	Using a programming language specified by the instructor, convert the pseudocode for a selected process to program code, incorporating at least three of the following structures, the need for which will be dictated by the assigned problem(s) and process(es). The resulting code design can be event-driven, object-oriented, or procedural.
	    a. Operations and functions (user-defined and/or library)
	    b. Repetition (loops)
	    c. Decision (if…else, case)
	    d. Recursion
	(TN Reading 3; TN Writing 4)		

10	Verify the correct operation of the resulting program code with several test cases:
	    a. All valid values
	    b. Error trapping of invalid values
	    c. Error trapping of invalid program operation
	    d. Troubleshooting/remedying program problems
	(TN Reading 3, 4; TN Writing 4)		

11	Compile the necessary documentation to understand the nature of a computer programming problem and the customer/client specifications for the request and summarize in an informational text. This will include evidence of the scope of the problem, its attendant input and output information, the required system processing, and the software specifications involved. (TN Reading 1, 2, 3; TN Writing 2, 8)		• **FBLA:** Help Desk, Computer Problem Solving
12	Analyze a given problem and develop a coherent strategy in the form of a project plan to meet the customer/client's need. The plan will include, but will not be limited to, defining the project scope as addressed by the problem documentation, identifying software development and implementation issues, timeline and benchmarks for design, and addressing issues associated with software maintenance and life cycle. (TN Reading 1, 2; TN Writing 2, 8)		• **FBLA:** Computer Problem Solving

13	In the software development process, articulate the nature of the program designs by creating documentation that addresses topics including but not limited to:
	    a. The procedural, object-oriented, event-driven, or other nature of the various portions of the resulting application
	    b. The data structures used for inputs, outputs, and internal manipulations
	    c. The algorithms and guiding formulas used
	    d. Constraints on accurate operation and results
	    e. Modular designs that enable portability
	    f. Interface details that permit ready maintenance and upkeep
	(TN Reading 6; TN Writing 2, 6)		

14	Apply principles of quality assurance during application development to certify bug tracking, audit trails, testing results, and other quality considerations. Annotate each quality assurance task with evidence from best practices endorsed by industry or research. (TN Reading 3, 6; TN Writing 7)		

15	Document the security risks associated with new applications and evaluate the severity of the risk involved in each, including but not limited to:
	   a. Identifying threats to information systems facilities, data communications systems, and other applications
	   b. Adhering to federal and state legislation pertaining to computer crime, fraud, and abuse
	   c. Providing means for preserving confidentiality and encryption of sensitive data
	   d. Detailing steps to recover from routine errors or catastrophic failures, such as might be caused by a malicious computer virus
	(TN Reading 8; TN Writing 1, 4)		

ALL	**CAN BE USED WITH ALL/MOST STANDARDS**	• **SkillsUSA**: Career Pathways Showcase, Job Skills Demonstration A, Job Skills Demonstration O, Prepared Speech, Extemporaneous Speaking, Chapter Display, Computer Programming • **TSA**: Software Development, Video Game Design	• **FCCLA**: Illustrated Talk, Career Investigation, Chapter in Review Display, Chapter in Review Portfolio